

# Anisotropic Mesh Adaptation for Image Representation and Scaling

Xianping Li, *Member, IEEE*

**Abstract**—Triangular meshes have gained much interest in image representation and have been widely used in image processing. This paper introduces a particular anisotropic mesh adaptation (AMA) method to image representation and image scaling. The AMA method is based on metric-specified mesh adaptation and finite element interpolation for Delaunay triangles. An initial triangular mesh is generated based on the desired number of points, and the mesh is then adapted according to a metric tensor that controls the size, shape and orientation of the triangles. Finally, the image is reconstructed from the mesh using finite element interpolation. The method is denoted as AMA-L if linear interpolation is used and AMA-Q if quadratic interpolation is used in the reconstruction, respectively. Different than many other methods, the AMA method starts with a triangular mesh directly and then adapts the mesh to represent the image. This AMA method has clear mathematical framework and can improve computational efficiency and accuracy in image processing. The AMA representation method provides comparable results with other content-based adaptive schemes but requires lower computational cost, and the AMA scaling method uses finite element interpolation and is comparable to vectorization.

**Index Terms**—Image representation, image scaling, anisotropic mesh adaptation, finite element interpolation, triangular mesh

## I. INTRODUCTION

Triangular meshes have recently received considerable interest in adaptive sampling for image representation [1]–[16]. For example, Terzopoulos and Vasilescu [2] sample an image at a reduced rate and then reconstruct it by concentrating the nodes of the mesh at regions where the image values change rapidly (high-gradient region). They develop adaptive meshes with a feedback procedure that automatically adjusts spring parameters according to the observations made at the nodes to which they are attached, and use a Gaussian convolution of the Hessian for the adaptive image reconstruction. Ramponi and Carrato [5] define a sample skewness parameter and use a multi-resolution approach to obtain a grid with an almost uniform sample density along the edges and no samples in areas with constant or linearly changing grey level. Yang *et al.* [8] argue that small (in area) elements are needed in image region where the second directional derivative is large and introduce the error diffusion (ED) scheme. They construct a feature map based on the Hessian of the image function first, then use Floyd-Steinberg dithering scheme to generate sample points, and finally use Delaunay triangulation to connect the nodes into a mesh. Demaret and Iske [11] propose the greedy-point removal (GPR) scheme that first constructs a triangular

mesh using all the image points and then removes the sample points that yield smallest error repeatedly. Courchesne *et al.* [12] use the Hessian matrix based on the gray level of MRI images as a metric tensor to adapt the triangular mesh for 3D reconstruction of human trunk. Bougleux, Peyre and Cohen [14] develop a progressive geodesic meshing algorithm that exploits the anisotropy of images through a farthest point sampling strategy and forces the anisotropic Delaunay triangles to follow the geometry of the image. They demonstrate the advantages of anisotropic triangular approximation over isotropic triangular approximation. Sarkis and Diepold [15] use binary space partitions in combination with three clustering schemes and reconstruct the image using the planar equations defined from the three nodes of each triangle. Adams [16] proposes the GPRFS method based on the GPR scheme by replacing the initial triangular mesh of all image points with a subset of the points which are selected using ED method.

Most of the adaptive sampling methods are “content-based” that use the information from the image values such as edges, textures, and gradients or Hessian. Corresponding mesh strategies and interpolations methods are used to reconstruct the image. The GPR method provides high quality meshes but requires significant computational cost. On the other hand, the ED method reduces computational costs but provides lower quality meshes. The GPRFS method tries to find a balance between mesh quality and computational cost by combining the advantages of GPR and ED methods. It is worth mentioning that adaptive meshes are used in [2], [12] and [14] to represent the image directly without taking sample points, and our anisotropic mesh adaptation (AMA) method to be introduced in this paper follows the same procedure.

Anisotropic mesh adaptation (AMA) has been successfully applied to improve computational efficiency and accuracy when solving partial differential equations (PDEs) [17]–[22]. In this paper, we introduce a particular AMA method that uses the metric tensors defined in [23] to image representation and scaling. We will show that AMA method not only has clear mathematical framework but also provides good results in image representation and scaling. The outline of the paper is as follows. Firstly, a brief introduction of the AMA method is given in section II. The details of the method can be found in [23] and [24]. It is briefly introduced here for reader’s convenience. Then in III, the advantage of AMA method in image representation is discussed and the results are compared with other methods. In section IV, the AMA method is applied to image scaling. Finally, some conclusions and comments are given in section V.

The author is with Department of Mathematics and Statistics, University of Missouri-Kansas City, Kansas City, MO 64110, U.S.A. (lixianp@umkc.edu)

## II. ANISOTROPIC MESH ADAPTATION (AMA) METHOD

Different adaptive sampling method and mesh strategies have been applied in image representation by other researchers as summarized in Section I. In this section, we introduce a particular “anisotropic mesh adaptation” (AMA) method. AMA takes the  $M$ -uniform mesh approach, with which an adaptive mesh is generated as a uniform mesh in the metric specified by a tensor  $M = M(x)$ . The metric tensor  $M$  is required to be strictly positive definite and it determines the size, shape and orientation of the triangular elements [23]. The metric tensor  $M$  is computed based on the original image data. Once a metric tensor is specified, the free C++ code BAMG (Bidimensional Anisotropic Mesh Generator) developed by Hecht [25] is used to generate the corresponding Delaunay-type triangular mesh.

Firstly, we introduce some notations and the conditions for  $M$ -uniform meshes. Let  $\Omega$  be the spatial domain,  $K$  be any triangular element in a simplicial mesh  $\mathcal{T}_h$ , and  $\hat{K}$  to be the reference element that is equilateral and unitary in area. Let  $F_K$  be the affine mapping from  $\hat{K}$  to  $K$  (see Fig. 1).

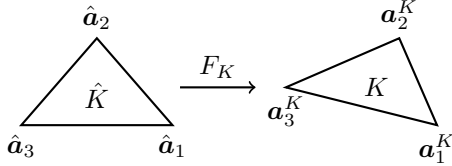


Fig. 1. Sketch of coordinate transformations from  $\hat{K}$  to  $K$ . Here,  $\hat{K}$  is the reference element and  $F_K$  is the affine mapping from  $\hat{K}$  to element  $K$ .

An  $M$ -uniform mesh for a given metric tensor  $M = M(x)$  satisfies the following two conditions [23]:

$$\rho_K |K| = \frac{\sigma_h}{N_e}, \quad \forall K \in \mathcal{T}_h \quad (1)$$

$$\frac{1}{d} \text{tr}((F'_K)^T M_K F'_K) = \det((F'_K)^T M_K F'_K)^{\frac{1}{d}} \quad (2)$$

where  $|K|$  is the area of the element  $K$ ,  $N_e$  is the number of mesh elements,  $d$  is the dimension,  $F'_K$  is the Jacobian matrix of  $F_K$ ,

$$M_K = \frac{1}{|K|} \int_K M(x) dx, \quad \rho_K = \sqrt{\det(M_K)} \quad (3)$$

and

$$\sigma_h = \sum_{K \in \mathcal{T}_h} \rho_K |K|. \quad (4)$$

Condition (1) is called *the equidistribution condition* and determines the size of  $K$  from  $\rho_K$ . Condition (2) is called *the alignment condition* and characterizes the shape and orientation of  $K$ .

Different metric tensor  $M$  will have different properties and features. For example, taking  $M$  as identity matrix provides a uniform mesh where all the triangles are of the same size and shape. There is no content-based adaptation for uniform mesh.

For isotropic mesh adaptation, a metric tensor  $M_{iso}$  is defined for any triangle element  $K$  as follows [26].

$$M_{iso,K} = \left(1 + \frac{1}{\alpha_h} \|H_K(u)\|_F\right) I \quad (5)$$

where  $I$  is the identity matrix of size  $2 \times 2$ ,  $H(u)$  is the Hessian of  $u$ ,  $H_K(u)$  denotes the value of  $H(u)$  at the center of element  $K$ ,  $\|\cdot\|_F$  is the Frobenius matrix norm, and  $\alpha_h$  is a regularization factor that is defined by

$$\alpha_h = \frac{1}{|\Omega|} \left( \sum_{K \in \mathcal{T}_h} |K| \cdot |H_K(u)| \right). \quad (6)$$

$M_{iso}$  provides isotropic mesh adaptation where all the triangles are of the same shape but may have different sizes.

For anisotropic mesh adaptation, we choose a metric tensor  $M_{aniso}$  that is based on minimization of a bound on the  $H^1$  semi-norm of linear interpolation error and is defined for any triangle element  $K$  as follows [26].

$$M_{aniso,K} = \rho_K \det \left( I + \frac{1}{\alpha_h} |H_K(u)| \right)^{-\frac{1}{2}} \left[ I + \frac{1}{\alpha_h} |H_K(u)| \right] \quad (7)$$

and

$$\rho_K = \left\| I + \frac{1}{\alpha_h} |H_K(u)| \right\|_F^{\frac{1}{2}} \det \left( I + \frac{1}{\alpha_h} |H_K(u)| \right)^{\frac{1}{4}} \quad (8)$$

where  $|H(u)| = \sqrt{H(u)^2}$ , and  $\alpha_h$  is defined implicitly through

$$\sum_{K \in \mathcal{T}_h} \rho_K |K| = 2|\Omega|. \quad (9)$$

Notice that  $M_{aniso}$  makes use of the Hessian of the image function and the corresponding mesh will concentrate more triangle elements in the region where image data changes significantly. The adaptation is anisotropic because the triangles in the mesh may have different size, shape and orientation.

It is worth mentioning that anisotropic meshes have advantages over isotropic meshes in terms of computational efficiency and accuracy. Fig. 2 shows both isotropic and anisotropic meshes that represent the following function that simulates the interaction of a boundary layer with an oblique shock wave [26]

$$f(x, y) = \tanh(60y) - \tanh(60(x - y) - 30)$$

for  $(x, y) \in (0, 1) \times (0, 1)$ .

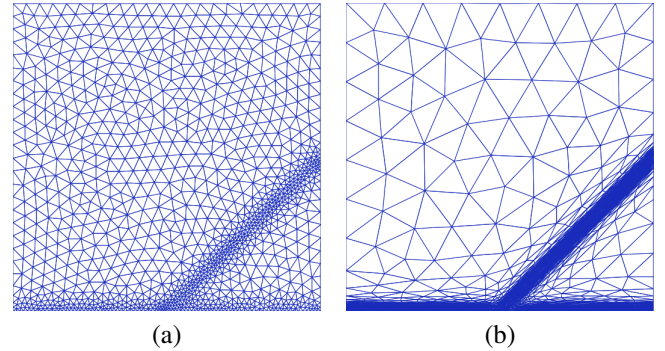


Fig. 2. Triangular meshes that represent the function  $f(x, y) = \tanh(60y) - \tanh(60(x - y) - 30)$  for  $(x, y) \in (0, 1) \times (0, 1)$ : (a) isotropic mesh based on  $M_{iso}$  containing 1248 vertices and 2326 triangles; (b) anisotropic mesh based on  $M_{aniso}$  containing 1231 vertices and 2323 triangles.

Fig. 2 clearly demonstrates that anisotropic mesh adaptation  $M_{aniso}$  presents the feature better than isotropic mesh

adaptation  $M_{iso}$ . Therefore, we only consider anisotropic mesh adaptation (AMA) and focus on  $M_{aniso}$  in this paper. Other metric tensors can be considered in the similar way.

### III. AMA REPRESENTATION

We consider an image as a function  $f$  that is defined on a set  $\Lambda$  of points on domain  $\Omega = [0, 1] \times [0, 1]$ . Let  $S$  denote the set of sample points and  $SD$  denote the sample density that is defined as  $SD = |S|/|\Lambda|$  where  $|\cdot|$  is the cardinality of the set. The peak-signal-to-noise-ratio (PSNR) in decibels (dB) is calculated as follows [16].

$$\text{PSNR} = 20 \log_{10} \left( \frac{M}{d} \right) \quad (10)$$

where

$$d = \left( \frac{1}{|\Lambda|} \sum_{i \in \Lambda} |\hat{f}(i) - f(i)|^2 \right)^{\frac{1}{2}}, \quad M = 2^\rho - 1 \quad (11)$$

$\hat{f}$  is the reconstructed image from the triangular mesh, and  $\rho$  is the sample precision in bits/sample. Larger value of PSNR indicates better mesh quality or representation.

Adams proposes the GPRFS method in [16] that is based on the GPR scheme while replacing the initial triangular mesh of all image points with a subset  $S_0$  of the points. It reduces the computational cost significantly comparing to GPR. However, the choice of  $S_0$  is critical for this method. In his paper, Adams employs the ED method to select  $S_0$  and proposed the GPRFS-ED method. It is identical to GPR scheme when choosing  $S_0 = \Lambda$  (the set of all points). Adams also proposed the GPRFS-MED, a modified light version of GPRFS-ED for lower computational cost. He has tested a few images and compared the results of GPRFS-ED and GPRFS-MED with ED and GPR schemes. Among the tested images, two of them (Lena and Peppers) with pixel resolution  $512 \times 512$  are available from the provided reference [27]. In order to compare with Adams' results, the two images, Lena and Peppers, are converted using Matlab function `rgb2gray()` from colorful TIFF files to gray scale JPG files with sample precision  $\rho = 8$ . Other images are not chosen for comparison due to the uncertainty of the source.

Now we apply the AMA method based on  $M_{aniso}$  for image representation, in which a triangular mesh with fewer points is used to represent the original image  $f$ . We use the free C++ code BAMG [25] to generate and adapt mesh according to  $M_{aniso}$ . Given the original image  $f$ , BAMG is used to generate a relatively small initial triangular mesh  $\mathcal{T}_0$  based on the geometry file and the desired number of points. The choice and effect of initial mesh will be discussed later. Then the values on the vertices are interpolated from  $f$ , and the metric tensor  $M_{aniso}$  is computed for each triangle in the initial mesh  $\mathcal{T}_0$ . With the computed metric tensor  $M_{aniso}$ , BAMG employs five local minimization tools, edge suppression, vertex suppression, vertex addition, edge swapping, and vertex reallocation (barycentering step) to generate the desired anisotropic mesh. The new mesh based on  $M_{aniso}$  is quasi- $M$ -uniform and denoted by  $\mathcal{T}_M$ . This procedure can be repeated

2 to 5 times in order to obtain a final mesh with good quality. Finally, the image is reconstructed from the final mesh using finite element interpolation (linear or quadratic).

More specifically, the AMA representation method consists of the following four steps.

Step 1: Generate an initial mesh based on the geometric file and the desired sample density.

Step 2: Assign values to mesh vertices from  $f$  using linear finite element interpolation and compute the metric tensor  $M_{aniso}$  on the mesh. No quadratic interpolation is needed for this step.

Step 3: Adapt the mesh to an  $M$ -uniform mesh according to  $M_{aniso}$ .

Step 4: Reconstruct the image using the  $M$ -uniform mesh with finite element interpolation for triangles.

The above procedures are shown in Fig. 3 where Step 2 and Step 3 can be repeated 2 to 5 times. We consider two methods based on different interpolation in the reconstruction (the last step). One is called AMA-L method that is based on linear finite element interpolation, and the other is called AMA-Q method that is based on quadratic finite element interpolation. For reader's convenience, the finite element interpolation methods for triangles are briefly described in Appendix A.

Remark for Step 1: The initial mesh  $\mathcal{T}_0$  can be adjusted using the parameters in the geometric file and the desired sample density  $SD$ . In our computations, we choose  $|S_0| = SD \times |\Lambda|$ , and the initial mesh has much fewer points than the image pixels for small  $SD$ . If all the image points are taken ( $SD = 100\%$ ), the initial values on the vertices will be directly from image data without any interpolation error.

Remark for Step 4: The sample density is the same for AMA-L and AMA-Q. Although AMA-Q method uses both the vertices and the middle points of the sides for quadratic interpolation, only the three vertices are needed as the sample points. The coordinates of middle points in the mesh and the corresponding function values are computed during the quadratic interpolation. In fact, we adapt the mesh to represent the image directly without taking sample points first, and the mesh is the same for AMA-L and AMA-Q.

Fig. 4 shows the initial images of Lena and Peppers with pixel resolution  $512 \times 512$ . They are obtained from [27] and converted to gray scale JPG files using Matlab function `rgb2gray()` with sample precision  $\rho = 8$ . The results will be compared to the ones in Adam's paper [16].

Fig. 5 shows the meshes and sample points of the image Lena at sample density of 3% after iterating Step 2 and Step 3 different times using AMA-L (in Step 4). The peak-signal-to-noise-ratio (PSNR) is 28.98 after one iteration, and is 29.97 after two iterations. After 5 iterations, the PSNR is 30.37 and the results are shown in Fig. 6.

More iterations of Step 2 and Step 3 will adapt the mesh to satisfy the  $M$ -uniform conditions (1) and (2) better, hence the representation will be better. In our computations, 5 iterations are performed to provide the final  $M$ -uniform mesh, unless otherwise stated.

Fig. 6 shows the representation of the image Lena at sample density of 3% and the corresponding triangular meshes

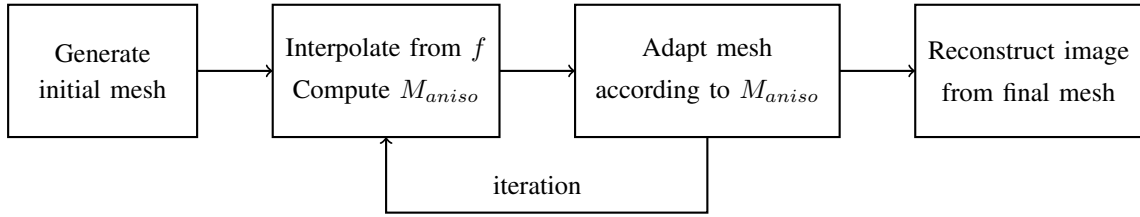


Fig. 3. Procedures for AMA representation method

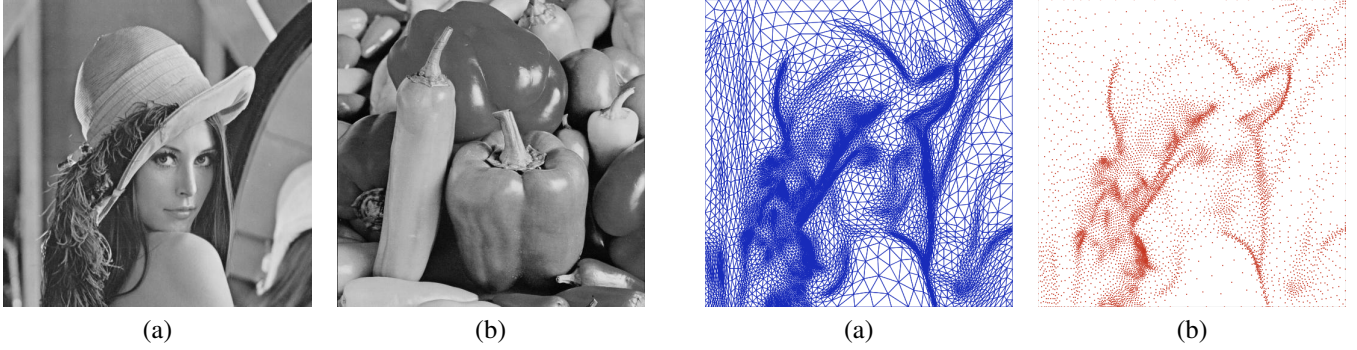


Fig. 4. Images from USC-SIPI Image Database [27] with pixel resolution  $512 \times 512$ : (a) Lena; (b) Peppers

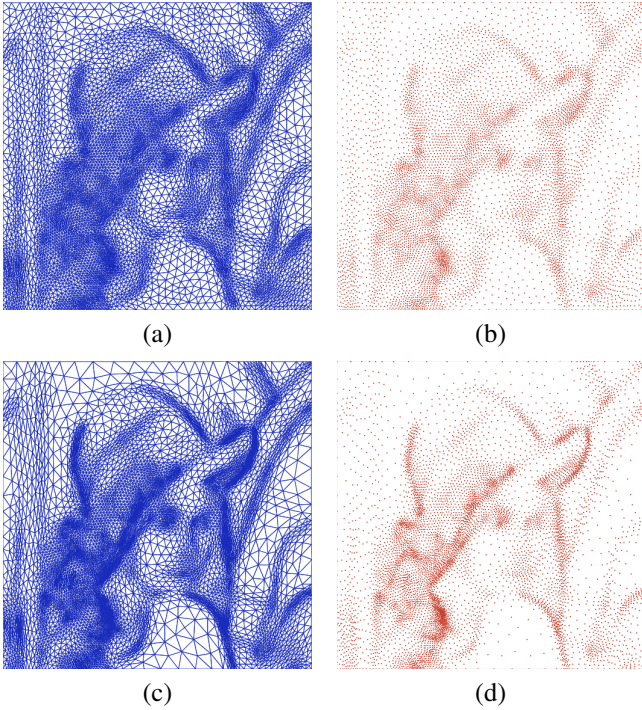


Fig. 5. Meshes and sample points of image Lena at sample density of 3%: (a) triangular mesh after one iteration; (b) sample points from the mesh in (a); (c) triangular mesh after two iterations; (d) sample points from the mesh in (c).

obtained according to the metric tensor  $M_{aniso}$ . The PSNR is 30.37 using AMA-L and is 34.56 using AMA-Q. Fig. 7 shows the representation of the image Peppers, also at sample density of 3%. The PSNR is 30.53 using AMA-L and is 34.72 using AMA-Q. Notice that the sample points and triangular meshes are the same for AMA-L and AMA-Q.



Fig. 6. Representations of image Lena at sample density of 3%: (a) triangular mesh; (b) sample points from the mesh in (a); (c) Representation using AMA-L, PSNR=30.37; (d) Representation using AMA-Q, PSNR=34.56.

As can be seen from Fig. 6 and Fig. 7, the triangular meshes preserve key features of the original image by concentrating more triangle elements around the edges and texture regions. The AMA-Q representation is better than the AMA-L representation because more information are preserved due to the more accurate quadratic interpolation.

The mesh qualities using AMA-L and AMA-Q are compared with ED, GPRFS-ED, and GPRFS-MED in Table I. For AMA-L and AMA-Q, the initial meshes are generated by BAMG from the number of points determined by sample density. That is,  $|S_0| = SD \times |\Lambda|$ , which is much smaller than  $|\Lambda|$ . The computational cost is significantly reduced by choosing small  $|S_0|$ .

As can be seen from Table I, the AMA-L method is comparable to GPRFS-MED, both performing much better than the ED method. The AMA-Q method performs even better than GPRFS-ED method.

Notice that the results in Table I for AMA-L and AMA-Q are obtained using  $|S_0|$  that is much smaller than  $|\Lambda|$ . When choosing  $S_0 = \Lambda$ , the corresponding PSNR will be slightly higher. For example, at sample density of 3.0%, for image



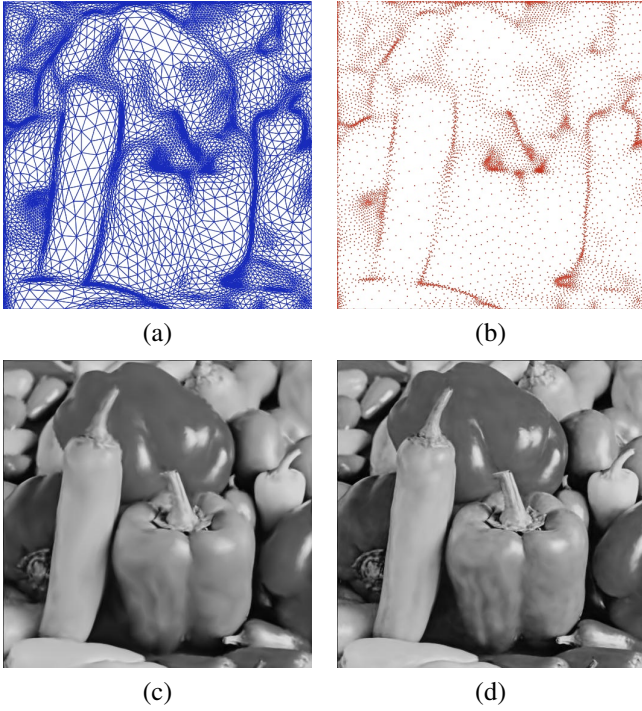


Fig. 7. Representations of image Peppers at sample density of 3%: (a) triangular mesh; (b) sample points from the mesh in (a); (c) Representation using AMA-L, PSNR=30.53; (d) Representation using AMA-Q, PSNR=34.72.

TABLE I  
COMPARISON OF MESH QUALITY OBTAINED WITH VARIOUS METHODS

Image	Sample Density (%)	PSNR (dB)				
		ED	GPRFS-MED*	GPRFS-ED	AMA-L	AMA-Q
Lena	1.0	21.13	23.37 (0.63)	29.21	25.62	29.49
	2.0	25.83	27.43 (1.37)	32.00	28.76	32.87
	3.0	28.05	29.50 (2.07)	33.55	30.37	34.56
	4.0	29.58	30.87 (2.85)	34.60	31.31	35.45
Peppers	1.0	21.35	23.34 (0.64)	30.20	25.64	29.67
	2.0	26.08	27.45 (1.34)	32.70	28.89	33.16
	3.0	28.16	29.62 (1.98)	33.82	30.53	34.72
	4.0	29.84	31.39 (2.65)	34.55	32.04	35.79

\*The numbers in parentheses are the actual sampling densities in percent.

Lena, PSNR=30.60 using AMA-L and PSNR=35.00 using AMA-Q; and for image Peppers, PSNR=30.92 using AMA-L and PSNR=34.95 using AMA-Q. The increments are all less than 0.5. Since choosing  $S_0 = \Lambda$  does not improve the mesh quality significantly, it is reasonable to choose smaller  $|S_0|$  in order to reduce the computational cost.

Fig. 8 shows the PSNR at different sample densities using both AMA-L and AMA-Q for image Lena. When the sample density reaches 100%, the PSNR can only reach to 44.30 using AMA-L, and can reach to 51.52 using AMA-Q. The PSNR is increasing with sample density at approximately the rate of  $1/8$ . The results for image Peppers show the similar behavior. Notice that there is no significant improvement in mesh quality when sample density is larger than 3%. Therefore, it is

reasonable to represent images at sample density of 3%.

#### IV. AMA SCALING

Next, we apply AMA method for image scaling. Image scaling is very important due to the different sizes of displays such as TVs, computer monitors, smartphones, and medical image devices. There already exist many algorithms for image scaling such as Nearest-neighbor interpolation, Bilinear interpolation, Bicubic interpolation, Lanczos interpolation/resampling, New edge-directed interpolation (NEDI) [28], Supersampling and Vectorization. Among the schemes, Lanczos interpolation is widely used due to its advantages of higher subpixel accuracy, better preservation of small-scale structures, and less generation of aliasing artifacts. It has been used for the Preview software in OS X system since Snow Leopard (version 10.6) for image size adjustment [29].

With the AMA representation method described in Section III, the AMA scaling method follows naturally. It is the same as the AMA representation method except in the last step. In the last step of AMA scaling, the final image resolution is different than the original image and can be defined by a scale factor  $sf$ . The desired image is reconstructed from the triangular representation using finite element interpolation. If  $sf < 1$ , the reconstruction gives a downsampled image; and if  $sf > 1$ , the reconstruction provides an upscaled image. For image upscaling, we choose  $S_0 = \Lambda$  in order to preserve as much information as possible, although smaller initial mesh can be chosen to reduce computational cost. For image downscaling, a smaller initial mesh is chosen based on sample density of 3%.

Fig. 9 shows the result of downscaling image Peppers from  $512 \times 512$  by a scale factor  $sf = 1/8$ . The triangular mesh is generated by BAMG according to  $M_{aniso}$  at the sample density of 3%. Then the downsampled image is reconstructed by linear finite element interpolation (AMA-L). The result is comparable to the one obtained using the Preview 6.0.1 software in OS X system, but has the advantage of smaller file size (in JPG format).

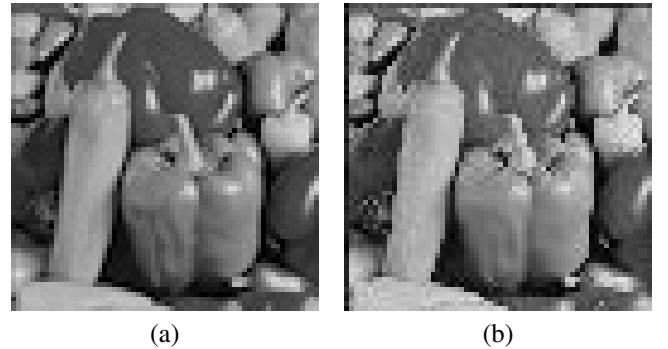


Fig. 9. Downsampling of image Peppers from  $512 \times 512$  to  $64 \times 64$ : (a) by Preview 6.0.1, image file is 4.7 KB; (b) by AMA-L, image file is 1.7 KB.

Fig. 10 shows the result of upscaling image Peppers from  $64 \times 64$  shown in Fig. 9(a) by a scale factor  $sf = 8$ . The triangular mesh is generated by BAMG according to  $M_{aniso}$  at the sample density of 100% (that is,  $S_0 = \Lambda$ ). Then

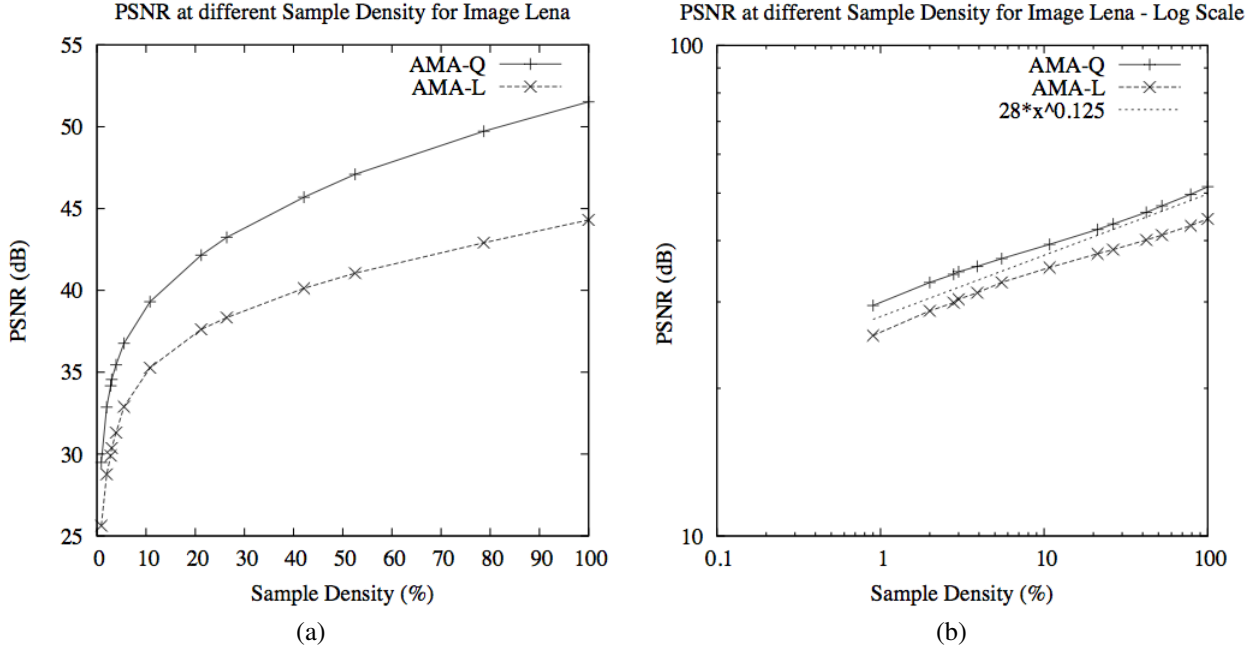


Fig. 8. PSNR at different sample densities for image Lena: (a) regular scale; (b) log-log scale.

the upscaled image is constructed by linear finite element interpolation (AMA-L) method. The result is comparable to the one obtained using the Preview software, and the image file (in JPE format) has smaller size.

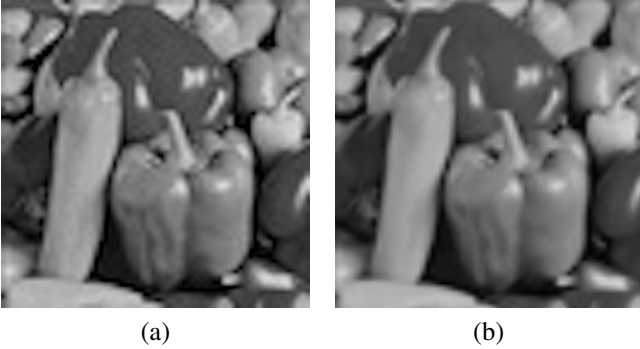


Fig. 10. Upscaling of image Peppers from  $64 \times 64$  to  $512 \times 512$ : (a) by Preview 6.0.1, image file is 37.1 KB; (b) by AMA-L, image file is 16.5 KB.

Next, we use AMA-L for both downscaling and upscaling. We first downscale the image Thank from  $256 \times 256$  to  $32 \times 32$ , then upscale the obtained  $32 \times 32$  image back to  $256 \times 256$ . The same procedure is repeated using Preview software for comparison, and the results are shown in Fig. 11. Although Preview seems to preserve the curvature better, the results from AMA-L preserve the black-white contrast better.

Lastly, we upscale the image Wiki with pixel resolution of  $109 \times 40$  that is obtained from Wikipedia ([http://en.wikipedia.org/wiki/Image\\_scaling](http://en.wikipedia.org/wiki/Image_scaling)). The file was converted to gray scale JPG file using the Matlab function `rgb2gray()`, and some noise was introduced during the conversion. The AMA upscaling has the same property as vectorization and can be easily applied to any scale factor without deteriorating the quality. The results are shown in Fig.

12, where (d) shows the upscaled image with  $sf = 32$  that has the same quality as the upscaled image in (c) with  $sf = 8$ .

In order to compute the PSNR, we first upscale the image Wiki by a scale factor  $sf = 8$  and then downscale the obtained image by a factor of  $sf = 1/8$ . The final image has the same pixel resolution with the initial image and the PSNR is computed. The results are shown in Fig. 13. Using Preview software, PSNR=22.19, while using AMA-L, PSNR=26.61. Consistently using AMA-L for both upscaling and downscaling provides better results than the Preview software.

## V. CONCLUSIONS AND COMMENTS

Adaptive sampling has become popular in image representation and scaling, among which triangular meshes have gained much interest. In this paper, we have introduced a particular anisotropic mesh adaptation (AMA) method to image representation and scaling. The AMA method follows the metric-specified  $M$ -uniform mesh approach, and uses the metric tensor  $M_{aniso}$  defined in (7) to control the triangular mesh. The image is represented by an  $M$ -uniform mesh and can be reconstructed to different pixel resolutions using finite element interpolation. The results from AMA representation and scaling are comparable to other methods, and choosing a small initial mesh significantly reduces the computational cost.

AMA method has clear mathematic framework and provides flexibility due to different choices of the metric tensor. Although we have only focused on the metric tensor  $M_{aniso}$  in this paper, other metric tensors can be considered in the similar way. Furthermore, AMA can also be applied to solve PDE-based image processing problems such as image smoothing and edge enhancement. Further application of AMA to other image processing problems will be investigated.

Thank  
you

(a)

Thank Thank  
you you

(b)

(c)

Thank Thank  
you you

(d)

(e)

Fig. 11. Scaling of image Thank: (a) initial image  $256 \times 256$ ; (b) downscaling from (a) to  $32 \times 32$  by Preview; (c) upscaling from (b) to  $256 \times 256$  by Preview; (d) downscaling from (a) to  $32 \times 32$  by AMA-L; (e) upscaling from (d) to  $256 \times 256$  by AMA-L.

## APPENDIX

### FINITE ELEMENT INTERPOLATION FOR TRIANGLES

This Appendix gives a brief introduction to the finite element interpolation for triangles: linear interpolation and quadratic interpolation. We consider a triangular element  $K$  and an isosceles right triangle  $\hat{K}$  as the reference element, as shown in Fig. 14. The vertices of  $K$  are denoted as  $\mathbf{a}_1$ ,  $\mathbf{a}_2$  and  $\mathbf{a}_3$ , and the corresponding middle points of the sides are denoted by  $\mathbf{a}_4$ ,  $\mathbf{a}_5$  and  $\mathbf{a}_6$ . The vertices of the reference element  $\hat{K}$  are located at  $\hat{\mathbf{a}}_1(0, 0)$ ,  $\hat{\mathbf{a}}_2(1, 0)$ , and  $\hat{\mathbf{a}}_3(0, 1)$ . The middle points in  $\hat{K}$  are located at  $\hat{\mathbf{a}}_4(0.5, 0)$ ,  $\hat{\mathbf{a}}_5(0.5, 0.5)$ , and  $\hat{\mathbf{a}}_6(0, 0.5)$ .

Denote the coordinates of the vertices of  $K$  as  $\mathbf{a}_1(x_1, y_1)$ ,  $\mathbf{a}_2(x_2, y_2)$ , and  $\mathbf{a}_3(x_3, y_3)$ . The corresponding function values are denoted by  $f_1 = f(x_1, y_1)$ ,  $f_2 = f(x_2, y_2)$  and  $f_3 = f(x_3, y_3)$ . For any point  $\mathbf{a}(x, y)$  in the element  $K$ , the corresponding point  $\hat{\mathbf{a}}(\xi, \eta)$  in the reference element  $\hat{K}$

Wiki

(a)

Wiki

(b)

Wiki

(c)

Wiki

(d)

Fig. 12. Upscaling of image Wiki: (a) initial image,  $109 \times 40$ , from [http://en.wikipedia.org/wiki/Image\\_scaling](http://en.wikipedia.org/wiki/Image_scaling); (b) by Preview with  $sf = 8$ ; (c) by AMA-L with  $sf = 8$ ; (d) by AMA-L with  $sf = 32$ .

Wiki Wiki

(a)

(b)

Fig. 13. First upscale with  $sf = 8$  then downscale with  $sf = 1/8$  for image Wiki: (a) by Preview, PSNR=22.19; (b) by AMA-L, PSNR=26.61.

is given by

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix}^{-1} \times \begin{bmatrix} x - x_1 \\ y - y_1 \end{bmatrix}. \quad (12)$$

For linear interpolation, only the values at three vertices are needed, and the function value is interpolated as follows.

$$f(x, y) = \sum_{i=1}^3 f_i \cdot C_i(x, y), \text{ or } f(\xi, \eta) = \sum_{i=1}^3 f_i \cdot N_i(\xi, \eta) \quad (13)$$

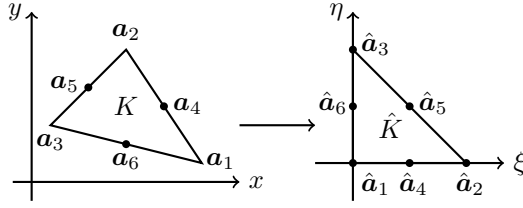


Fig. 14. Sketch of triangular element  $K$  and its reference element  $\hat{K}$ . Here,  $\hat{K}$  is an isosceles right triangle with vertices  $\hat{a}_1(0, 0)$ ,  $\hat{a}_2(1, 0)$ , and  $\hat{a}_3(0, 1)$ .

where  $C_i(x, y)$  and  $N_i(\xi, \eta)$  are the basis functions at  $\mathbf{a}_i$  and  $\hat{\mathbf{a}}_i$ , respectively, in the corresponding coordinate system. Specifically, in the reference element  $\hat{K}$ ,

$$N_1(\xi, \eta) = 1 - \xi - \eta; \quad N_2(\xi, \eta) = \xi; \quad N_3(\xi, \eta) = \eta. \quad (14)$$

For quadratic interpolation, the middle points of the sides are needed. Denote the corresponding function values at middle points as  $f_4$ ,  $f_5$  and  $f_6$ . Then the function value at any point  $(\xi, \eta)$  in  $\hat{K}$  is interpolated as follows.

$$f(\xi, \eta) = \sum_{i=1}^6 f_i \cdot N_i(\xi, \eta) \quad (15)$$

where

$$\begin{aligned} N_1 &= (1 - \xi - \eta)(1 - 2\xi - 2\eta); & N_2 &= \xi(2\xi - 1); \\ N_3 &= \eta(2\eta - 1); & N_4 &= 4\xi(1 - \xi - \eta); \\ N_5 &= 4\xi\eta; & N_6 &= 4\eta(1 - \xi - \eta). \end{aligned} \quad (16)$$

#### ACKNOWLEDGMENT

The author would like to thank the USC-SIPI Image Database and the Wikipedia website for providing some of the images used in this paper.

#### REFERENCES

- [1] N. Dyn, D. Levin, and S. Rippa, "Data dependent triangulations for piecewise linear interpolant," *IMA J. Numer. Anal.*, vol. 10, pp. 137–154, 1990.
- [2] D. Terzopoulos and M. Vasilescu, "Sampling and reconstruction with adaptive meshes," *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR'91)*, pp. 70–75, 1991.
- [3] F. Divoine, M. Antonini, J.-M. Chassery, and M. Barlaud, "Fractal image compression based on delaunay triangulation and vector quantization," *IEEE Trans. Image Process.*, vol. 5, no. 2, pp. 338–346, 1996.
- [4] H. Edelsbrunner, "Triangulations and meshes in computational geometry," *Acta Numerica*, pp. 133–213, 2000.
- [5] G. Ramponi and S. Carrato, "An adaptive irregular sampling algorithm and its application to image coding," *Image Vis. Computing*, vol. 19, pp. 451–460, 2001.
- [6] X. Yu, B. Morse, and T. Sederberg, "Image reconstruction using data-dependent triangulation," *IEEE Comput. Graph. Appl.*, vol. 21, no. 3, pp. 62–68, 2001.
- [7] J. Brankov, Y. Yang, and N. Galatsanos, "Image restoration using content-adaptive mesh modeling," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, pp. 997–1000, 2003.
- [8] Y. Yang, M. Wernick, and J. Brankov, "A fast approach for accurate content-adaptive mesh generation," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 866–881, 2003.
- [9] D. Su and P. Willis, "Image interpolation by pixel-level data-dependent triangulation," *Computer Graphics Forum*, vol. 23, no. 2, pp. 189–201, 2004.
- [10] J. Brankov, Y. Yang, and M. Wernick, "Tomographic image reconstruction based on a content-adaptive mesh model," *IEEE Trans. Med. Imaging*, vol. 23, no. 2, pp. 202–212, 2004.
- [11] L. Demaret and A. Iske, "Adaptive image approximation by linear splines over locally optimal delaunay triangulations," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 281–284, 2006.
- [12] O. Courchesne, F. Guibault, J. Dompierre, and F. Cheriet, "Adaptive mesh generation of mri images for 3d reconstruction of human trunk," *ICIAI 2007*, vol. LNCS 4633, pp. 1040–1051, 2007.
- [13] M. Sarkis and K. Diepold, "A fast solution to the approximation of 3-d scattered point data from stereo images using triangular meshes," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, vol. Pittsburgh, PA, pp. 235–241, 2007.
- [14] S. Bougleux, G. Peyré, and L. Cohen, "Image compression with geodesic anisotropic triangulations," *Proc. ICCV'09*, pp. 2343–2348, 2009.
- [15] M. Sarkis and K. Diepold, "Content adaptive mesh representation of images using binary space partitions," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 1069–1079, 2009.
- [16] M. Adams, "A flexible content-adaptive mesh-generation strategy for image representation," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2414–2427, 2011.
- [17] P. Frey and F. Alauzet, "Anisotropic mesh adaptation for cfd computations," *Comput. Methods. Appl. Mech. Engrg.*, vol. 194, pp. 5068–5082, 2005.
- [18] W. Huang and X. Li, "An anisotropic mesh adaptation method for the finite element solution of variational problems," *Fin. Elem. Anal. Des.*, vol. 46, pp. 61–73, 2010.
- [19] X. Li and W. Huang, "An anisotropic mesh adaptation method for the finite element solution of heterogeneous anisotropic diffusion problems," *J. Comput. Phys.*, vol. 229, pp. 8072–8094, 2010.
- [20] A. van Dam and P. Zegeling, "Balanced monitoring of flow phenomena in moving mesh methods," *Commun. Comput. Phys.*, vol. 7, pp. 138–170, 2010.
- [21] X. Li and W. Huang, "Maximum principle for the finite element solution of time-dependent anisotropic diffusion problems," *Numer. Meth. PDEs*, vol. 29, pp. 1963–1985, 2013.
- [22] J. Wackers, G. Deng, A. Leroyer, P. Queutey, and M. Visonneau, "Adaptive grid refinement for hydrodynamic flows," *Computers & Fluids*, vol. 55, pp. 85–100, 2012.
- [23] W. Huang, "Mathematical principles of anisotropic mesh adaptation," *Comm. Comput. Phys.*, vol. 1, pp. 276–310, 2006.
- [24] W. Huang and R. Russell, *Adaptive Moving Mesh Methods*. New York: Springer, 2011.
- [25] F. Hecht, bidimensional Anisotropic Mesh Generator software, available at: <http://www.ann.jussieu.fr/hecht/ftp/bamg/bamg-v1.01.tar.gz>, 2010.
- [26] W. Huang, "Metric tensors for anisotropic mesh generation," *J. Comput. Phys.*, vol. 204, pp. 633–665, 2005.
- [27] USC-SIPI Image Database. 2010. [Online]. Available: <http://sipi.usc.edu/database>.
- [28] X. Li and M. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, 2001.
- [29] M. Editors and K. Turner, *Total Snow Leopard (Macworld Superguides)*. Mac Publishing, L.L.C., 2009, ISBN-13: 978-0982262146.



**Xianping Li** (M'13) received the B.E. and M.S. degrees in petroleum engineering from the China University of Petroleum-Beijing, in 2000 and 2002, respectively; the M.S. degree in chemical & petroleum engineering and Ph.D. degree in mathematics from the University of Kansas, Lawrence, KS, in 2010 and 2011, respectively.

From 2011 to 2013, he was a Visiting Assistant Professor in the Department of Mathematics at the University of Central Arkansas. He has developed the theoretical conditions for triangular meshes such

that the numerical approximations for anisotropic diffusion problems are free of non-physical solutions. Since 2013, he has been with the Department of Mathematics and Statistics, University of Missouri-Kansas City, Kansas City, MO, where he is currently an Assistant Professor. His research interests include numerical solutions for partial differential equations, finite element method, anisotropic mesh adaptation, and mathematical modeling and simulation in engineering.

Dr. Li is a member of Society for Industrial and Applied Mathematics, Society of Petroleum Engineers, and American Mathematical Society.